STEP & XML

White Paper

Technologies for Digital Product Data Integration

Issue 1.0 – June 2001

European Marine STEP Association

http://emsa.germanlloyd.org/

Executive summary

> "Standards are not strictly about encoding -- they are also about setting a level of acceptable interoperability between real programs." **[01]**

By comparison with the traditional pace of standards development activities, the meteoric rise of XML is spectacular. XML enables the transmission of data and, by being universal in appeal, apparently offers a compelling alternative to existing domain-specific standards for data interchange. STEP is the major standard of relevance to engineering enterprises that work with product data. This White Paper examines the relationship between STEP and XML, in particular contrasting the purposes and capabilities that these standards bring to engineering organisations. The conclusion is that STEP and XML are complementary technologies and are both necessary with appropriate recognition of the respective strengths and purposes of the two standards.

STEP is the "Standard for the Exchange of Product Model Data", which has formal status as an International Standard, namely ISO 10303. According to Part 1 of the standard, ISO 10303 specifies "a form for the unambiguous representation and exchange of computer-interpretable product data throughout the life of a product".

XML is the "Extensible Markup Language", which is a Recommendation of the World Wide Web Consortium (W3C) in the form of a base language (XML 1.0). The base language enables the structured representation of data. XML now also consists of a growing family of technologies, which include capabilities that support links between data, distributed hosting of data, transformation between different representations and richer specification of structure.

STEP enables three key capabilities: data exchange; data sharing; and data archiving. Meanwhile, under Internet-driven momentum, XML has risen, appearing to offer an alternative to STEP by providing such capabilities, with the additional benefit that low-to-no cost tools are available to support the implementation of XML technologies. The most prevalent of such tools are the browsers for exploring the World Wide Web.

XML is gaining a reputation as being the solution to limitations in the World Wide Web. During the to-date formative years of the Web, HyperText Markup Language (HTML) has been the core technology and has successfully delivered documents to human users of the Internet. However, XML offers a more flexible capability to deliver data from one computer application to another and onwards to the user community that usually encompasses a wide range of various requirements. In a way that is barely possible with HTML, one source XML file can be the basis for multiple forms of presentation of data to the end users, for example showing different levels of detail.

On the other hand, the many attractions of XML do not actually challenge the foundations of STEP, which specifies a comprehensive architecture to deliver information models that are complete, unambiguous and conformance testable. These models are independent of implementation technology and, thus, have the longevity to outlast any particular encoding mechanism, software application or database. These applications and databases cover a wide range of engineering disciplines that include computer-aided design (CAD), computer-aided manufacture (CAM) and configuration and asset management. XML is now a pressing example of an encoding mechanism that STEP can utilise.

In the face of increasing computerisation of engineering business processes, the extended engineering enterprise requires product data integration so as to achieve "faster, better, cheaper". However, this extended enterprise is a challenging environment in which to establish such integration [see **Figure 1**]. The key features of the environment include: a potentially multi-decade life-cycle for complex product systems such as a ship; a diverse collection of interacting engineering specialisms; and a functional and geographic dispersion of partner organisations. The ultimate objective is to provide the enterprise with an architecture that enables coherent data creation, data management and data delivery. XML and STEP both have various potential roles within this architecture.

STEP has established a common set of information requirements, which are key necessary ingredients in achieving integration of product data across the extended engineering enterprise. XML *per se* does not embody such requirements. If the XML community endeavours to establish independent specifications for product data integration without reference to existing technologies then the community will duplicate the effort of those who have worked on ISO 10303. However, in place of any rivalry, the capabilities of XML are an ideal opportunity for the wider implementation of STEP. Thus, the STEP community has developed a new part of the standard, Part 28, which enables XML as a representation mechanism for STEP-conformant schemata and instance data.

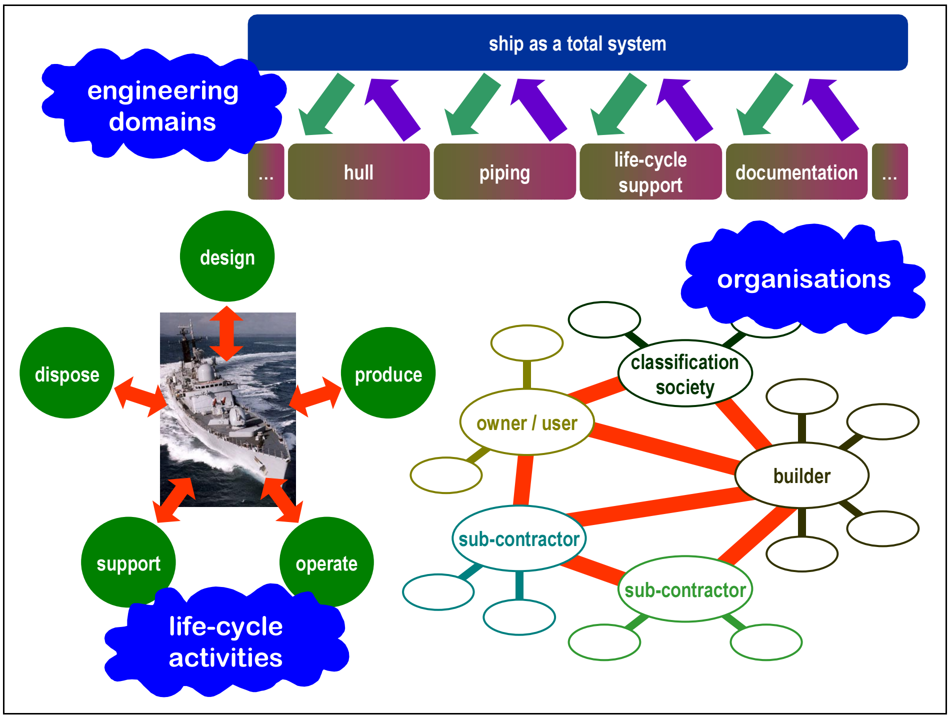


Figure 1 Features of the extended engineering enterprise

STEP versus XML is a false controversy. In order to achieve effective product data integration across the extended engineering enterprise, STEP and XML are complementary technologies and are both necessary with appropriate recognition of the respective strengths and purposes of the two standards. XML offers a powerful, generic mechanism for encoding of data. STEP is the broader framework that offers the prospect of interoperability between end-user software applications.

Contents

```
<manufacturer id = "ABCD">
          <manu_name>Smith & Jones Limited</manu_name>
          <manu_headquarters>Leeds, UK</manu_headquarters>
</manufacturer>

<part id = "1234">
          <part_number>56-7890</part_number>
          <part_description>Widget</part_description>
          <part_rel_date>19 January 1998</part_rel_date>
          <part.manufacturer>
                    <manufacturer_ref refid = "ABCD"/>
          </part.manufacturer>
</part>

<part id = "1235">
          <part_number>9A-431217890</part_number>
          <part_description>Widget flange</part_description>
          <part_rel_date>23 November 1997</part_rel_date>
          <part.manufacturer>
                    <manufacturer_ref refid = "ABCD"/>
          </part.manufacturer>
</part>
```

Glossary

AP	Application Protocol (member of ISO 10303 series of documents)
CAD	computer-aided design
CAM	computer-aided manufacture
DTD	Document Type Definition (SGML and XML)
EXPRESS	information modelling language for STEP (ISO 10303-11)
EXPRESS-X	mapping language for STEP (ISO 10303-14)
HTML	HyperText Markup Language (http://www.w3.org/MarkUp/)
IETM	interactive electronic technical manual
ISO	International Organization for Standardization (http://www.iso.ch/)
ISO 10303	"Industrial automation systems and integration -- Product data representation and exchange" (ISO standard; informal name STEP)
ISO 10303-1	"Overview and fundamental principles" (part of ISO 10303)
ISO 10303-11	"Description methods: The EXPRESS language reference manual" (part of ISO 10303)
ISO 10303-14	"Description methods: The EXPRESS-X language reference manual" (part of ISO 10303)
ISO 10303-21	"Implementation methods: Clear text encoding of the exchange structure" (part of ISO 10303)
ISO 10303-22	"Implementation methods: Standard data access interface" (part of ISO 10303)
ISO 10303-28	"Implementation methods: XML representation of EXPRESS schemas and data" (part of ISO 10303)
ISO 13584	"Industrial automation systems and integration -- Parts library" (ISO standard; informal name PLib)
ISO 15531	"Industrial automation systems and integration -- Industrial manufacturing management data" (ISO standard; informal name MANDATE)
ISO 15926	"Industrial automation systems and integration -- Integration of life-cycle data for oil and gas production facilities" (ISO standard; informal name Oil & Gas)
ISO 8879	"Information processing -- Text and office systems -- Standard Generalized Markup Language (SGML)" (ISO standard)
ISO/TC184/SC4	"Industrial data" (ISO technical subcommittee; http://www.tc184-sc4.org/)
MANDATE	informal name for ISO 15531
markup	use of tags to indicate the structure of data
MathML	Mathematical Markup Language (http://www.w3.org/Math/)
MatML	Materials Property Data Markup Language (http://www.ceramics.nist.gov/matml/matml.htm)

meta-data	data that characterise other data rather than objects in the world beyond the data set (for example, level of authorisation that is necessary to change part numbers for products)
Oil & Gas	informal name for ISO 15926
Part 21 file	informal name for data file that conforms to ISO 10303-21
PDML	Product Data Markup Language (http://www.pdml.org/)
PDTS	Preliminary Draft Technical Specification (ISO)
PDX	Product Definition eXchange (http://www.pdxstandard.org/)
PLib	informal name for ISO 13584 ("Parts Library")
RDF	Resource Description Framework (http://www.w3.org/RDF/)
RELAX	Regular Language description for XML (http://www.xml.gr.jp/relax/)
schema	collection of related definitions that form an information model (plural = “schemata”)
SDAI	Standard Data Access Interface (ISO 10303-22)
SGML	Standard Generalized Markup Language (ISO 8879)
SOAP	Simple Object Access Protocol (http://www.w3.org/TR/SOAP/) (XML)
STEP	informal name for ISO 10303 ("Standard for the Exchange of Product Model Data")
SVG	Scalable Vector Graphics (http://www.w3.org/Graphics/SVG/Overview.htm8)
tag	either of the pair of tokens that surround data content (for example, `<product>` or `</product>` in an XML document)
UCLP	Universal Commerce Language and Protocol (http://www.w3.org/TR/1999/NOTE-uclp-19990120/)
W3C	World Wide Web Consortium (http://www.w3.org/)
XInclude	XML Inclusions (http://www.w3.org/TR/xinclude/)
XLink	XML Linking Language (http://www.w3.org/XML/Linking.html)
XML	Extensible Markup Language (http://www.w3.org/XML/)
XPath	XML Path Language (http://www.w3.org/TR/xpath)
XPointer	XML Pointer Language (http://www.w3.org/XML/Linking.html)
XSL	Extensible Stylesheet Language (http://www.w3.org/Style/XSL/)
XSLT	XSL Transformations (http://www.w3.org/TR/xslt.html)

Notes on the context and audience for this White Paper

STEP & XML differ in so much and yet both have much to offer for the integration of product data.

After a few short years of development, XML has reached the attention of the average computer user. STEP is much older and perhaps appears to have no relevance to the vast bulk of information technology.

In the face of the seemingly relentless momentum of XML, some prophets of doom have found voice: "ISO procedures take too long"; "STEP is large and unwieldy"; and "XML will solve our problems". The purpose of this White Paper is to look beneath the hyperbole and identify the true relationships between two exciting technologies that are vital to the integration of digital product data.

Hold the hyperbole; XML is not "silver-bullet" technology and should be subject to proper matching against the requirements of the intended application.

In expectation that this White Paper will have an audience from both the XML and STEP communities, the technical focus of the document is critical to ensuring a balanced and shared understanding to a relevant depth. Thus, the next section contains a brief review of both STEP and XML. Where the reader is not sufficiently familiar with either or both of these subject areas, further details are available in the two Annexes.

The remainder of the White Paper will explore how XML and STEP are able to serve the requirements of the extended engineering enterprise in respect of product data integration.

A brief review of STEP and XML

In reviewing STEP and XML, the only reasonable conclusion is that these two technologies form a long list of contrasts. However, the starting point must be to clarify exactly what to call STEP and XML. The acronyms and origins are without dispute. STEP is the "Standard for the Exchange of Product Model Data". XML is the "Extensible Markup Language".

In principle, both "STEP" and "XML" are labels that refer to standards. STEP is the informal name for ISO 10303, "Industrial automation systems and integration -- Product data representation and exchange", under the responsibility of ISO technical subcommittee "Industrial data" (ISO/TC184/SC4). As a base language, XML 1.0 is a Recommendation of the World Wide Web Consortium (W3C).

Beyond the immediate existence of the standards, the word "technologies" is more suitable to capture the additional facets of both STEP and XML.

STEP is ISO 10303, which already consists of more than 60 published documents.

ISO 10303 is a multi-part standard that embodies an extensive architecture of components for information systems. The committee responsible for the standard continues to approve new projects to deliver further parts that increase the application domains of the standard. In addition, STEP becomes a product on the desk of the user through various software tools and functionality in applications, such as CAD or CAM.

The base language XML 1.0 is an application of Standard Generalized Markup Language (SGML), which is an International Standard (ISO 8879). However, the XML community has begun to deliver a host of capability that extends the features of the base XML. Thus, the label "XML" refers potentially to any one of four items:

XML is a standard but new requirements are driving the development of extensions to the base language.

- the base language, which in October 2000 became a Second Edition of the 1.0 Recommendation;
- extensions to the language, such as XML Inclusions (XInclude), XML Information Set and XML Fragment Interchange;
- support and implementation technologies and standards, such as Extensible Stylesheet Language (XSL), XML Query and XML Schema;
- industry and business initiatives to develop applications of XML, such as Mathematical Markup Language (MathML) and Metal XML.

On the basis of the above discussion, this White Paper must investigate more than the bare facts of the standards for the practical purposes of integrating product data. If the reader is unfamiliar with the background details of STEP [see **Annex A**] or XML [see **Annex B**] then this is the point at which to make a brief detour. Otherwise, the next section of this White Paper presents the background that has shaped the architectures of modern information technology systems.

An information science context for STEP and XML

What is digital product data?

In simple terms, digital product data is anything relating to product and existing in the computers of the product enterprise. However, the potential range of required product information is extensive and likely to exist in a large number of different formats.

Computer-based data is all-pervasive in the engineering enterprise.

Increasingly, digital product data will perhaps originate from onboard monitoring systems with delivery to end-user applications through automated collection. These systems can provide extensive information on the operational conditions of products. However, a great deal of information requires human intervention to provide either source data or configuration of the framework within which automation can operate. For example, in the case of stress analysis, the engineer will generate a model and then initiate the actual analysis. The process is probably iterative and a final set of results becomes a part of the design documentation.

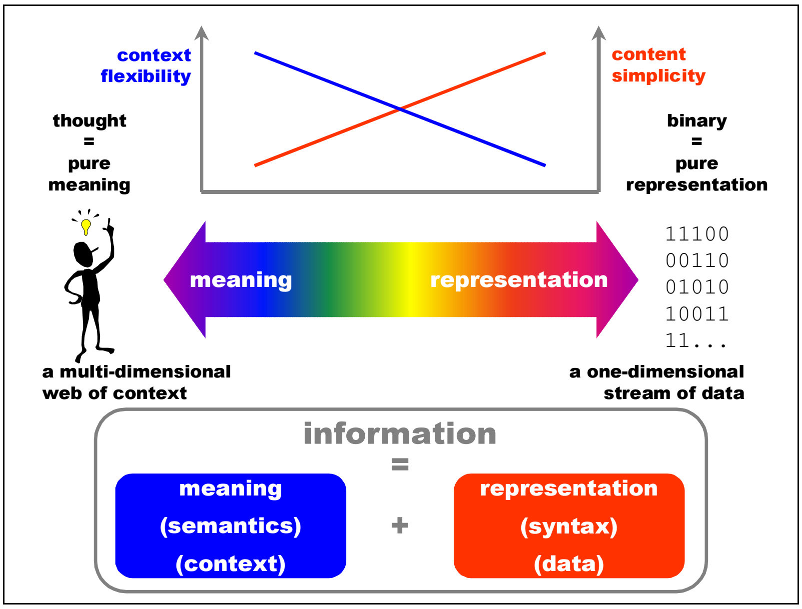
The role of data in information technology systems

Digital data is the primary fuel of the information technology revolution. However, as the distinct use of terms suggests, data is not the same as information. Instead, the context of the application software establishes the information content.

A typical example of product information is the engineering drawing with which the designer is familiar as a representation of product form. The purpose of the computer system is to deliver that representation so as to enable the designer to interact with and edit the drawing. However, the fundamental representation for the purposes of the computer will be the stream of binary data. Furthermore, this stream is probably specific to the particular operating system and version of the drawing software, such that the same drawing would have different binary representations. Such differences are a fundamental business issue for an extended product enterprise where communication of information is vital to successful operation of the enterprise.

Computers are very precise but that precision very easily prevents communication in a changing world; the human brain adapts with great flexibility.

The underlying nature of information is the existence of a continuum from pure meaning to pure representation [see **Figure 2**]. Digital product data is just a stream of binary states ("0" and "1"). The rich intended meaning arises from the context in which the user creates and accesses the data.



Note: Experience is the missing dimension that creates knowledge out of information.

Figure 2 The information continuum

Information is the combination of both data and context.

The computer is dependent on binary data because the hardware is capable of applying the rules of binary mathematics to the data consistently and accurately at great speeds. All further capabilities of the computer are human inventions relying on the foundations of binary data. The resultant architecture of a computer includes software components such as the operating system and programming language compilers. These components ensure that the user does not

The computer efficiently processes "1"s and "0"s but a rich architecture is necessary to deliver information to the senses of the user.

have to think in terms of "0" and "1" [see **Figure 3**]. Furthermore, the user increasingly is able to work with consistent frameworks despite changes in the fundamental components because the modularity of components eases the consequences of changes in the architecture. These frameworks include programming languages, such as Java, markup languages, such as HTML, and even the consistent graphical interface for word processors on different hardware platforms.

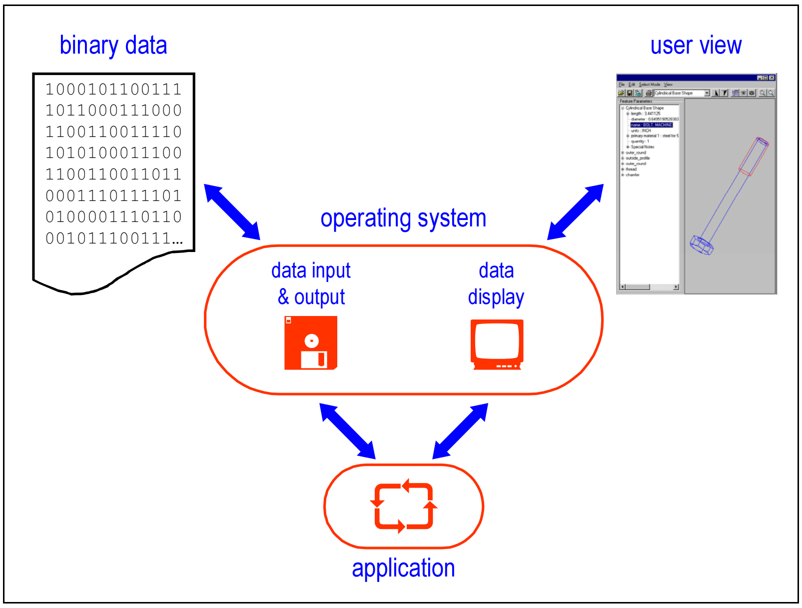


Figure 3 The user is isolated from the binary data that drives digital computers

The human mind and senses have mechanisms and capabilities that still far outstrip the general capabilities of the computer. Thus, people are able to produce and respond to representations such as engineering drawings with seemingly little effort.

The architecture of an application will either be open to public scrutiny or else a closed, proprietary solution.

In creating computer applications, the challenge is to produce the context in which rich representations become the medium for user interaction. The software industry continues to add to the wide diversity of such applications. However, the approach to development determines some fundamental features of the application architecture. Without external pressure, the typical approach to development will produce an inextricable binding of the context with the representation [see **Figure 4**]. Furthermore, the architecture will almost certainly consist of just two context layers; these layers are the stepping stones that allow the user to access the intended meaning of the digital data.

A closed application architecture poses commercial risk. For instance, the developer of the software will perhaps cease to trade. Furthermore, often the data will outlive the software in usefulness. A ship or aircraft has related product data that will have a life of decades; any given version of application software is unlikely to last more than a few years.

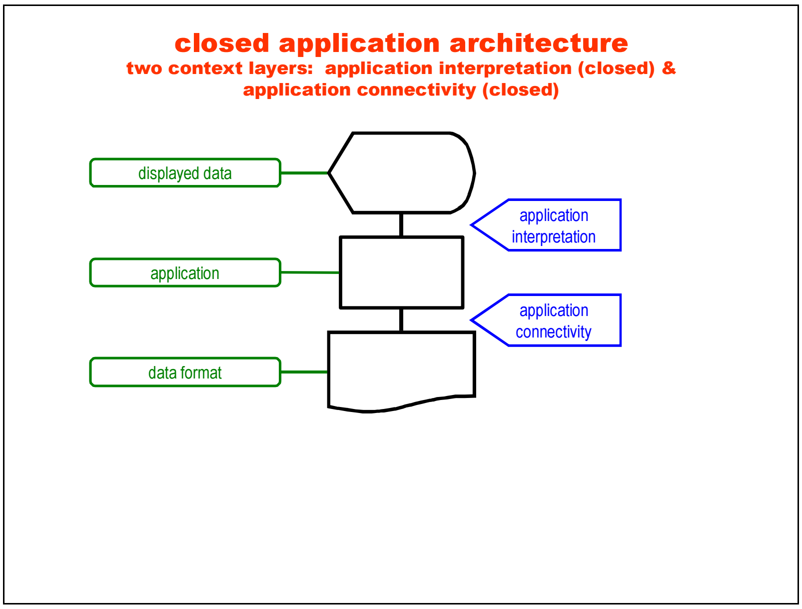


Figure 4 Proprietary, closed architecture application software

When clever people produce clever software the technology is probably powerful but more than this, in general, businesses require communication between open software so as to support the synthesis of data to produce new information.

The response to commercial risk has been for software users to demand the deployment of open standards [see **Figure 5**].

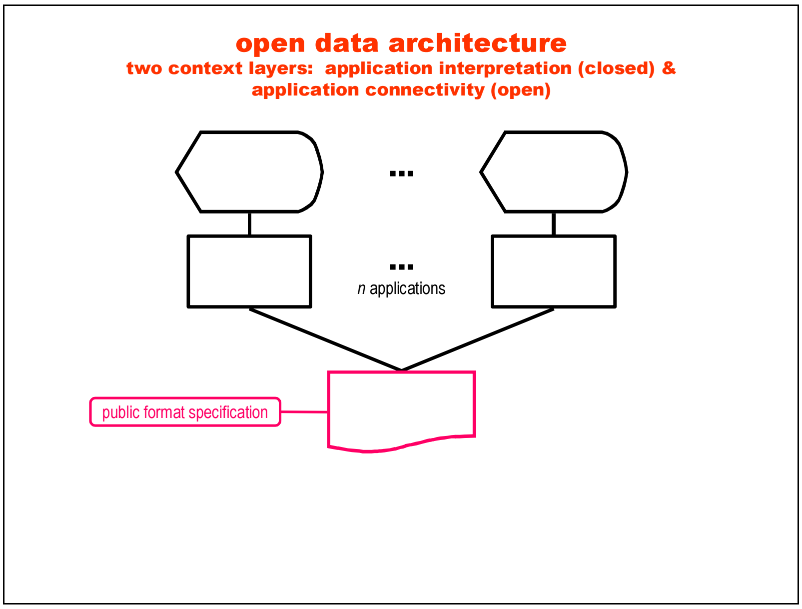


Figure 5 An open data architecture for application software

Some proprietary data formats are used for open exchange. For instance, the AutoCAD format (DXF) is a typical option on CAD systems. However, the owner of the format can change the specification at any time.

On a first review, the open data architecture appears to address the fundamental sources of commercial risk. However, by using only two context layers, the architecture has inherent inflexibility. The standardisation effort emphasises the information representation rather than the information meaning. Greater long-term viability arises from using a multi-layer architecture [see **Figure 6**]. The conceptual model is the mechanism by which the standard specifies meaning. Such a multi-layer architecture can consist of multiple implementation methods, further reducing commercial risk. As technology evolves, the standard remains valid in terms of the specification of information meaning. Furthermore, the implementation can rely on architectural components that are useful for other applications. Thus, economy of scale becomes possible.

A standard data format is useful but an information architecture is more powerful.

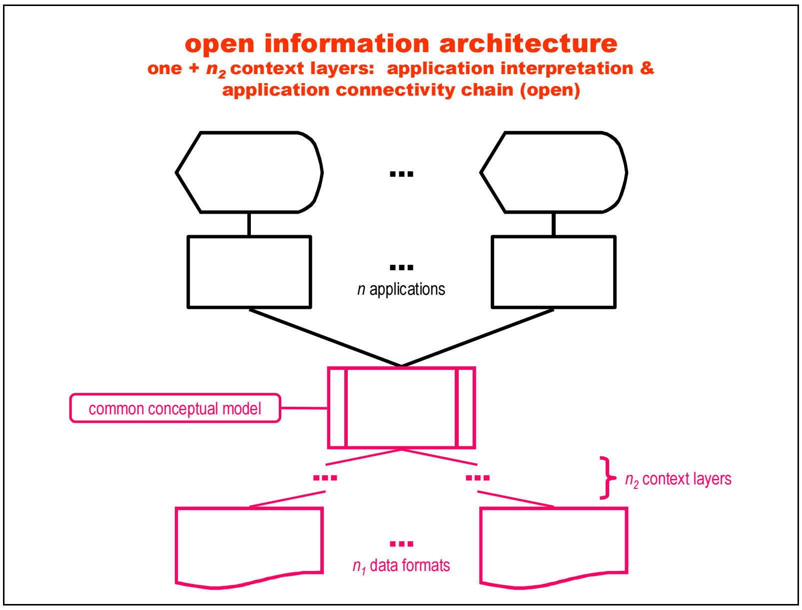


Figure 6 An open information architecture for application software

The discussion above is relevant to this White Paper because STEP and XML sit at particular points in the information continuum. The potential role of each standard arises from the capability to serve the requirements for open information architectures in modern software applications.

The environment for modern information management systems

The price / performance of hardware no longer constrains the deployment of information technology.

The information technology revolution has had many facets but one particularly important feature has been the interaction between hardware capability and software functionality. This interaction has been the foundation for the changes between generations of system architecture [see **Figure 7**].

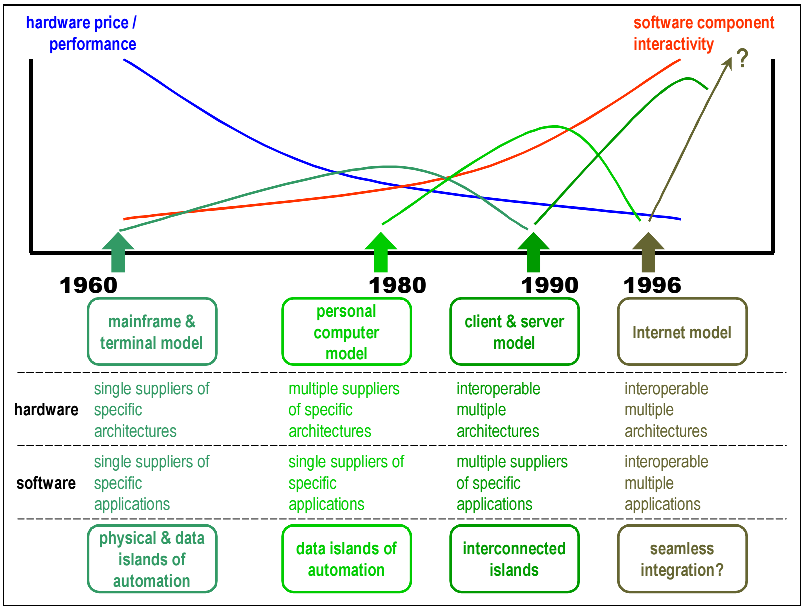


Figure 7 The changing paradigms of information technology systems

The seamless integration of systems requires compatibility between the underpinning conceptual information models.

The Internet model is still evolving. Many details of this model remain the subject of debate. For instance, different major players in the market have taken different views on the future roles of end-user computers and the consequent requirements for hardware capability and software configurations[1]. However, **Figure 7** is neutral with respect to such issues and instead concentrates on the fundamental features of the architectures.

An open information architecture is the only reasonable basis for the integration of product data.

With the foundation of extensive interoperability in both hardware and software, the Internet model offers the Holy Grail of integration of information management across the extended enterprise. XML is undoubtedly wrapped up in this revolution and is the focus of software development for many vendors. In turn, customers face strong commercial pressures to choose the in-vogue technology; especially when much browser technology is available at low-to-no cost.

The discussion above has laid out the landscape into which to place the main theme of this White Paper, namely the integration of product data.

[1] Typically, the crux of this debate is on whether end users will operate personal computers or thin clients.

An infrastructure for product data integration

The business context

In the modern industrial context, digital product data is the foundation for the integrated product life-cycle. Several business drivers create the pressure for this integration. These drivers include the increase in:

Modern products are complex, owners demand lower life-time costs and government legislation imposes accountability.

- functional complexity of product;
- emphasis on total cost of ownership;
- requirements to be accountable for safety issues and long-term environmental impact.

These business drivers extend the virtual enterprise that the product data exist to serve. The enterprise has three dimensions:

The extended enterprise encompasses diverse technical domains, changing processes over time and inter-dependent organisations.

- engineering domain. Many different domains contribute to the engineering of a product. Furthermore, any complex product is a system, which also extends to items beyond the main physical product. These items include documentation and the maintenance support system. Best practice in systems engineering requires iterative interaction between engineering domains [see **Figure 8**];
- life-cycle activity. The complete product life-cycle comprises five key activities [see **Figure 9**]. Furthermore, the life-cycle will probably last decades in the case of major assets such as ships and aircraft;
- organisation. The extended virtual enterprise comprises the different organisations that effect the life-cycle [see **Figure 10**].

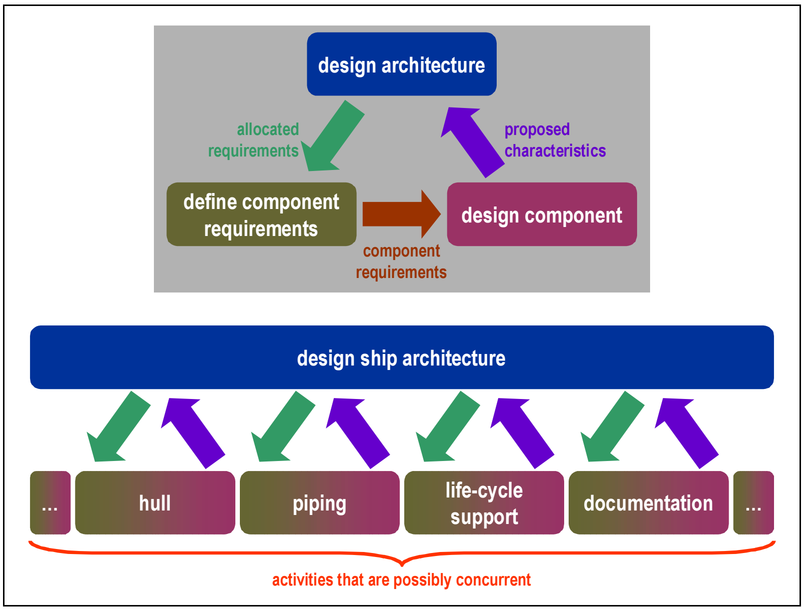
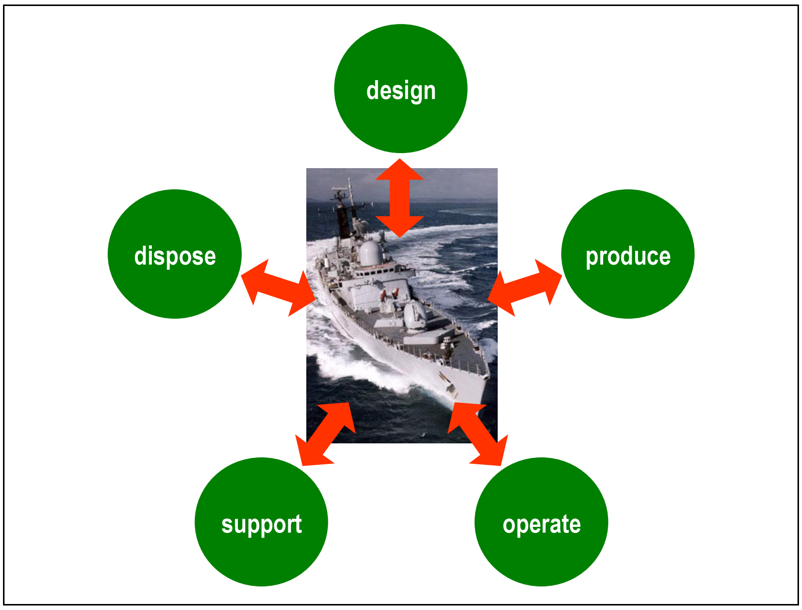


Figure 8 Examples of engineering input to the product design process

Best practice in systems engineering maximises and balances concurrent input from technical domains to achieve optimum results.



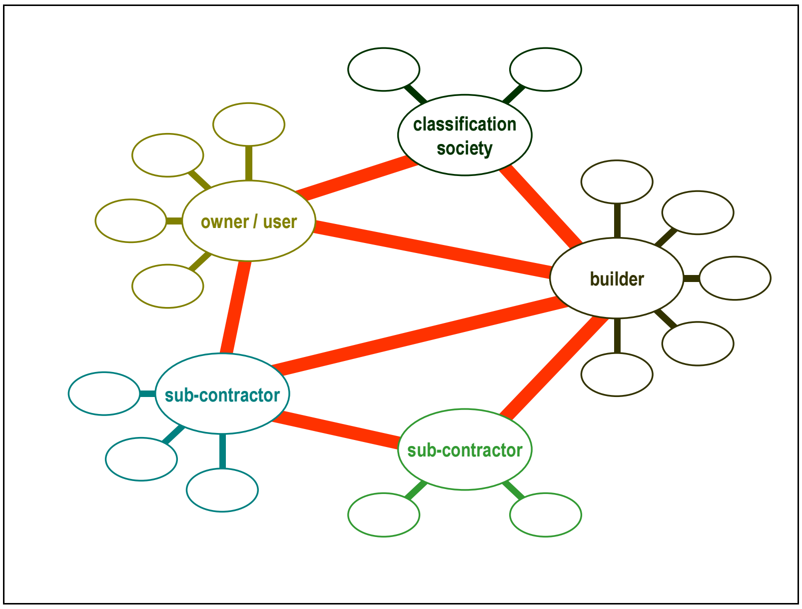
Note: further details are available in **[02]**.

Figure 9 The activities of the product life-cycle

A major asset does not have a linear life-cycle. Product improvements are frequent and engage all activities on an on-going basis.

Throughout the extended enterprise, engineers and others require access to the product data. Such access is for different purposes, in multiple organisations and throughout a potentially long life-cycle. Traditionally, the STEP community has seen the requirement in terms of a triplet that is "exchange, sharing & archiving" of product data.

Integration is all three but also includes enabling and managing links between different forms of data.



Note: the smaller ellipses represent departments and other sub-divisions of the individual organisations.

Figure 10 An example for the ship life-cycle of the organisational dimension of the extended enterprise

Extended enterprises build various types of major asset, such as ships, aircraft, process plant and transport systems.

An important business objective for effective application of product data is the mantra of "create once, use many". Any item of data should have a single point of origin, with no subsequent duplicated manual input. Quality and increased efficiency arise from electronic transmission of the data item to as many individuals who have a need to process the item.

The NASA teams shared data but did not share information.

Although the theory will perhaps appear trite, the lesson of the Mars Climate Orbiter is salutary; two teams within NASA used different units of measure and nobody realised until after the disaster [see **Figure 11**]. Complete and unambiguous product data is the panacea for such problems. STEP and XML are not technologies to establish the physical capability to exchange and share the data. The technologies for connectivity are the networking hardware and the communication protocols of the Internet. However, STEP and XML are technologies to establish the common basis on which to understand the data that flow across the connections.

Figure 11 Whoops! Now what did the physics teacher say about always using units …

Numbers without units are data without context.

The information requirements

On the basis of the above-described business context, the integration of product data encompasses a wide range of types of information within the extended engineering enterprise. The ultimate requirement is to capture and manage the knowledge that is the life-blood of the successful modern enterprise. Currently, most organisations manage data in structures with few of the links that ensure a sufficiently rich structure to reflect fully the underlying knowledge. The potential source of such links exists in the minds of people, who will eventually leave and access to the knowledge becomes lost.

Both coarse-grained and fine-grained data are necessary for the purposes of product engineering, operation and support.

In more specific terms, product data will take on a multitude of forms that include both coarse and fine grains. Any one of these forms is an "information object", an expression that is more generic than conventional terms such as "document", "file", "image" and so on. The following are examples of coarse-grained information objects:

- design drawings (for example, conceptual, developmental and product);
- requirements (user, system and so on);
- parts lists;
- engineering analysis models and results;
- technical data packages;
- technical documentation, such as interactive electronic technical manuals (IETM);

- further items such as video and product catalogues.

Fine-grained information objects arise as queries, messages and the like. Users seek specific information from large collections through queries. Partners in an enterprise inform each other of product life-cycle events through the exchange of messages.

Further characteristics of product data include the existence of inter-linkage and multiple levels of granularity. The inter-links in the data include examples such as the use in design drawings of part numbers as identification, where those part numbers exist in the product catalogue of a manufacturer. Reference data and dictionaries are important components of the information architecture into which links are necessary. Such components ensure maximum, accurate re-use of existing enterprise knowledge through a common, shared vocabulary. Multiple levels of granularity exist in, for instance, a technical data package that will perhaps contain a document that, in turn, will perhaps contain the results of a query.

Richly linked data are vital to successful knowledge-based engineering.

Finally, meta-data are an extremely important commodity in the extended enterprise. Meta-data are nothing more than data about data. However, the essence of information management is to control and distribute meta-data to enable more effective use of the underlying content data. The meta-data will exist at each of the multiple levels of granularity and different implementation technologies will possibly be necessary at different levels.

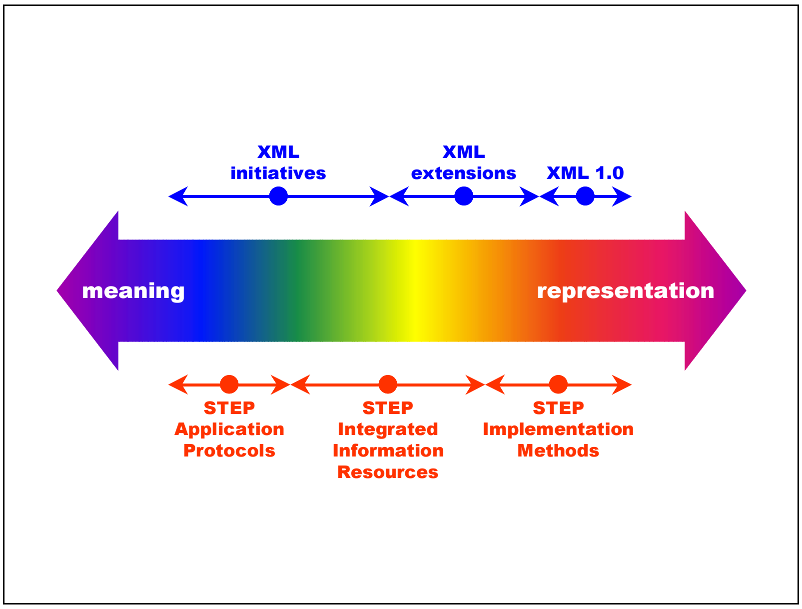
If the enterprise is able to establish inter-linking and other forms of structural richness within the information set then the ultimate goal of knowledge management is achievable.

Meta-data are the basis for management of information throughout the enterprise.

Information is likely to originate from some particular application for creation, serve a variety of purposes in other processing and presentation applications and pass through information management systems on the way. The engineering enterprise seeks a rationalisation of the technology to support these processes.

The roles of STEP and XML

By design, STEP is absolutely relevant to the requirements of product data integration. By contrast, XML is in a less clear-cut situation. However, the starting point is to consider the characteristics of the two standards in terms of the information continuum [see **Figure 12**].



Note: each range corresponds to potential data content; the label indicates the standard to which the particular data conform

Figure 12 STEP and XML in the information continuum

The STEP project set out to build an architecture; XML is an encoding (representation) mechanism that is now growing to face the bigger challenge.

The STEP architecture addresses the complete range of needs of the engineering enterprise. The Implementation Methods are the basis for flexible information systems; the Application Protocols provide for a communication of the information requirements of the users in the various specific engineering domains.

The base language of XML is not particular to any individual application domain. Thus, this base language does not specify the rich semantic content that is necessary to serve the needs of product data. Instead, such rich content is within the scope of XML initiatives.

The user can read XML tags and make a good guess as to the meaning of data but engineering enterprises demand greater robustness than subjective judgement.

Given that rich semantic content is necessary to ensure accurate, cost-effective and future-proof product data integration, XML technologies can provide support to such integration through either of two different approaches [see **Figure 13**]:

- establish an XML initiative that will serve the specific information requirements of product data integration;
- as part of STEP, create a new Implementation Method that uses XML as the representation.

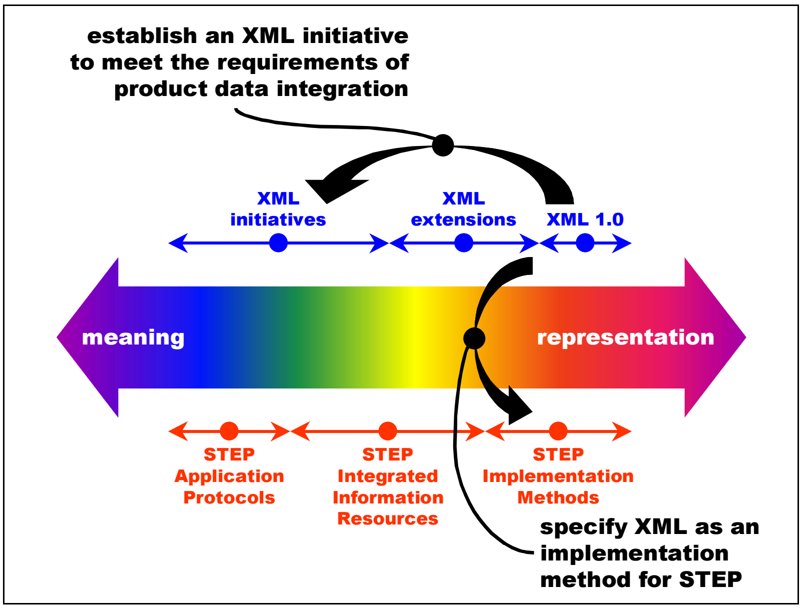


Figure 13 Serving the integration of product data through XML technologies

The XML community has the option to choose a blank piece of paper to start the design of a solution for product data integration.

In fact, different parties have investigated both approaches and are discovering various advantages.

XML initiatives for product data

Firstly, among the many XML initiatives that already exist **[03]**, the following are examples of initiatives that are in the sphere of product data:

"ML" projects have "X" appeal but "the devil is in the detail".

- DTD and schema for product data exchange **[04]**;
- Materials Property Data Markup Language (MatML) **[05]**;
- Metal XML **[06]**;
- Product Data Markup Language (PDML) **[07]**;
- Product Definition eXchange (PDX) **[08]**;
- Universal Commerce Language and Protocol (UCLP) **[09]**.

In terms of this White Paper, the most interesting of these initiatives is PDML **[10]**. The relevance of this particular initiative arises because the thinking behind STEP informed the development of PDML. The results of the initiative enable engineers to view product data across the Internet. PDML consists of a set of components, which form an architecture that closely mirrors the construction of STEP. A single, master Integration Schema contains many of the concepts of the STEP Integrated Generic Resources.

With respect to the potential for widespread adoption of the PDML initiative, one constraint is that the initiative was the result of a US Department of Defense programme. The specific objectives of the programme are not necessarily applicable to a broader user community.

PDML depends on STEP-based approaches to provide a robust framework for the integration of product data.

The other identified XML initiatives are from a variety of communities, both commercial and research. No single effort yet matches the size and scope of STEP. In the short term and for bounded purposes, the capabilities of XML are powerful motivators for the adoption of the language and associated technology. However, none of the initiatives has implemented an architecture that is as rich in domain engineering content or as flexible as STEP. For instance, in order to achieve the information modelling that is necessary to ensure the rigorous capture of information requirements, the PDML initiative was not able to deploy solely XML technologies and instead implemented the master models in the same form of the STEP models. This form is the EXPRESS language for data definition (ISO 10303-11).

XML technology is still maturing in respect of information modelling. With SGML as an origin for XML but primarily "a set of tools for managing documents" **[11]**, the Document Type Definition (DTD) has had a foundational role in the early history of XML. However, the DTD does not embody the principles that have grown within the domain of information analysis, whereby a rich conceptual model is the basis for information management systems.

The DTD in XML is a limited mechanism for specifying the structure of product data.

Although not the only proposal and this multiplicity is yet another issue, the XML Schema specification will deliver more flexibility than a DTD in respect of structure and the definition and validation of data types. However, XML Schema is unlikely to be as powerful as EXPRESS and this limitation will possibly leave the specification in a limbo between two more mature, well-understood and supported technologies in the form of the DTD and the EXPRESS schema **[12]**. In summary, EXPRESS is a capable modelling language, while DTD and XML Schema are validation languages.

The development of STEP to exploit XML technology

ISO/TC184/SC4 has recently achieved a formal milestone in the process of creating a new Implementation Method for STEP to make use of XML. On 26 October 2000, ISO began a three-month ballot of a preliminary Draft Technical Specification (PDTS) that is designated "ISO 10303-28" and entitled "Implementation methods: XML representation of EXPRESS schemas and data" **[13]**. In line with the usual convention within the ISO/TC184/SC4 community, this is "Part 28 of STEP".

Part 28 delivers the benefits of XML to STEP.

The title of STEP Part 28 is admirably descriptive of the functionality enabled by the standard. The challenge has been to align the content of the document with the fast-moving world of XML technology. In fact, this challenge has been the primary reason for seeking Technical

Specification status for Part 28. Under ISO rules, the document will have a shorter period before a required review (maximum of three years, rather than the five for a full International Standard).

The Part 21 file is not defunct but Part 28 will reach a broader audience than the one to date.

Without doubt, Part 28 has arisen from many different interests in the potential for XML to deliver a variety of benefits to STEP. Ultimately, time will tell as to which benefits prove to be the most significant but the following are all possible candidates:

- offer an alternative to the clear-text encoding of STEP data using ISO 10303-21 (the "Part 21 file"). As this alternative, XML is a technology that is more generic and allows developers to share code with access to other types of data in XML format;
- include the schema in the data file and ensure greater fidelity of the long-term archiving of the data;
- provide a new, common publication and distribution mechanism for the developed STEP schemata and, thus, reach a broader community who are not familiar with ISO/TC184/SC4;
- provide users of STEP data with access to the increasing range of XML tools that support useful functionality for data presentation and transformation and are often at viable prices for large-scale deployment within the extended engineering enterprise.

Tools for XML technology will flourish because the potential number of users offers a huge economy of scale.

XML offers a much larger potential for economy of scale than has ever existed in the STEP community. Although the specification of the STEP architecture includes all the necessary components to ensure robust information systems, the community has not been able to implement all the details of this specification. One particular example is the enforcement of the rule-based constraints that are a feature of EXPRESS schemata, for instance through use of the `WHERE` keyword [see **Figure 23**]. By deploying cheaper, generic XML technologies, the missing functionality is more likely to become available. These technologies will also offer added benefits from the complete range of additional XML capability. This capability includes the rich handling of links, allowing seamless presentation of referenced objects through a single interface.

The relative merits of the two standards are finely balanced in terms of the features of practical implementation. XML allows the initiative partners to define a set of tags that are more human-readable than the STEP Part 21 file. However, the Part 21 file is a rationalised view of the data. This rationalisation is the basis for software vendors to implement solutions through common components to handle STEP data and, thus, reduce costs and development times and increase quality. A further issue is that the XML tags increase the relative size of the exchange file. Although, technology for data compression is increasingly a component of the software environment, end-user applications must still process the file. In particular, complex geometric representations are likely to require very large files. The price / performance of computer hardware continues to fall and, thus, mitigates concerns about size.

Whether through an XML initiative or STEP Part 28, effective technologies for product data integration are now a realistic prospect and paint an exciting picture for the future of the extended enterprise. However, the implementation of these technologies poses various challenges. The following section of this White Paper explores the potential shape of the future and highlights some suitable approaches to facing the challenges.

The practicalities of implementing product data integration

Building a ship provides a typical example of the nature of an extended engineering enterprise.

The ship industry provides a perfect example for the requirements of through-life product data in an extended enterprise. The ship enterprise consists of the following members: the builder; sub-contractors (design and manufacturing); the classification society; the user; the owner; the support organisation; and other types of partner. The relevant STEP Application Protocols cover a broad range of domain requirements, through-out the life-cycle of a ship [see **Table 1**].

Table 1 The ship Application Protocols within ISO 10303

Note: Changes to AP227 have provided for the requirements of ship piping (originally the subject of a separate document, AP217)

Number	Title	Current status
215	Ship arrangements	Working Draft
216	Ship moulded forms	Committee Draft
218	Ship structures	Committee Draft
226	Ship mechanical systems	Committee Draft
227	Plant spatial configuration (2nd Edition)	Draft International Standard
234	Ship operational logs, records, and messages	Working Draft

Exchange, sharing and archiving should not occur as a free-for-all frenzy; management control is necessary to protect the well-being of the enterprise.

In an extended enterprise, three distinct functional phases become the focus of the information system solution. These phases are [see **Figure 14**]:

- data creation;
- data management; and
- data delivery.

These phases encompass the core capabilities of STEP (exchange, sharing and archiving) but in the broader enterprise context. As an example, although the basic operation will perhaps be to generate an exchange file for transfer of a ship moulded form from CAD application software at one partner into different software at another partner, the exchange should not occur in a vacuum. This exchange should be the subject of appropriate management procedures that will include, for instance, an audit trail to cover creation and distribution of data. Furthermore, a two-party "push-pull" approach limits the visibility of information to the broader community of the enterprise partners. By deploying a managed repository, the enterprise is able effect widespread access to up-to-date information.

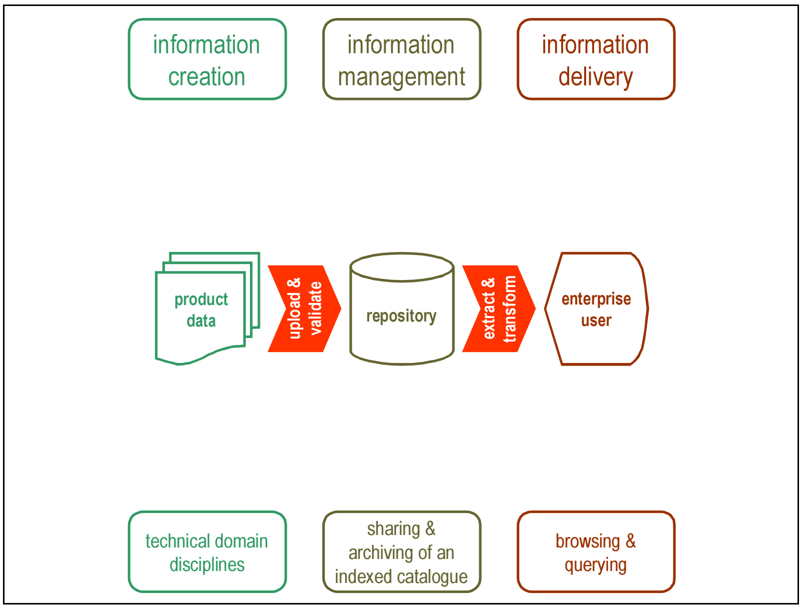


Figure 14 A system architecture for product data in the extended enterprise

The top-level view of the architecture hides an inter-operating blend of technologies.

An initial observation is that the data sources, intermediate connectivity mechanisms and delivery media are all potentially independent technologies. Some particular technology will be suitable for each aspect of the architecture. This suitability will depend on the benefits of that technology with respect to the particular extended enterprise. Appropriate interfaces will be necessary between the different technologies.

STEP Application Protocols provide a robust basis on which to create data for distribution to other end-user software.

In the phase of data creation [see **Figure 15**], STEP has an important role. In particular, the life-cycle activity of design is one where large quantities of detailed data are necessary to fuel the software applications that support the activity. ISO 10303 has an extensive capability in design and already enables data exchange among most major CAD applications. Other data creation will not be STEP enabled. Legacy standards and proprietary formats will still be applicable and the creation of documentation will possibly make use of SGML or XML.

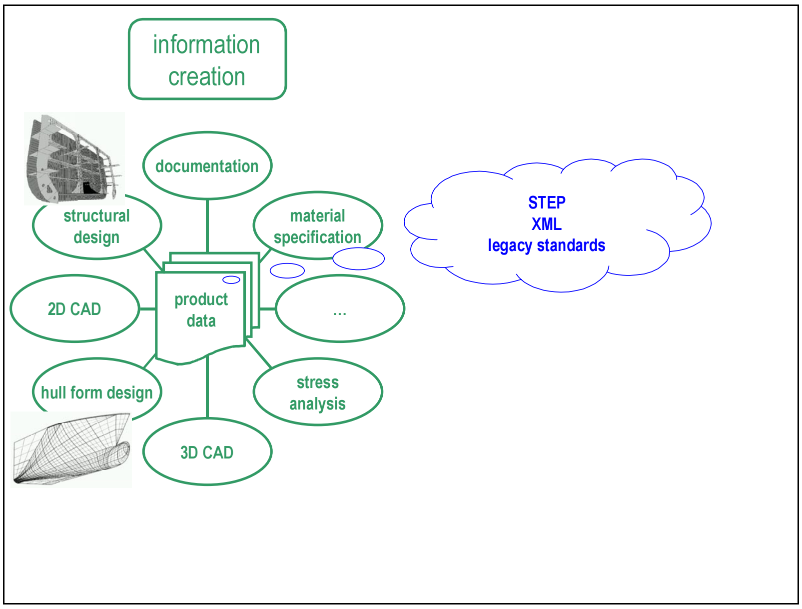


Figure 15 Examples of product data from a variety of sources

Data integrity is vital to the successful auditing of activities in the enterprise.

In order to populate the repository that is necessary for data management within the enterprise context [see **Figure 16**], the created data must undergo an upload process. An important component of the upload process is the validation of data. In the case of STEP and XML, the formal validation criteria are clear. STEP requires conformance to an EXPRESS schema and, for most current XML implementations, a DTD is the basis for validation.

The operation of the repository is subject to many different factors. The overall objective is to establish the basis on which to manage the data for the extended enterprise. In particular, meta-data are fundamental to the management tasks **[14]**. These meta-data are fine-grained and accessible to applications that interact with the repository. A large proportion of the content data can exist as coarse-grained information objects and the format of these objects can be appropriate to the content, which will perhaps range from documents to videos and from engineering drawings to parts lists. Without doubt, the repository will necessarily store data in STEP format.

In the corporate pyramid, only a few will want editing rights or fully detailed access to information. The many will want summary views.

The next link in the chain is to present the data for distribution. Just as the input to the repository is upload <u>and</u> validate, so the output consists of both extract and transform. The transform is necessary to deliver views of the data that correspond to the needs of different types of user. For instance, a reviewer will probably only require access to a layout drawing of a design rather than the complete CAD model. XML offers XSLT as an important potential technology for transformation but EXPRESS-X is also an option.

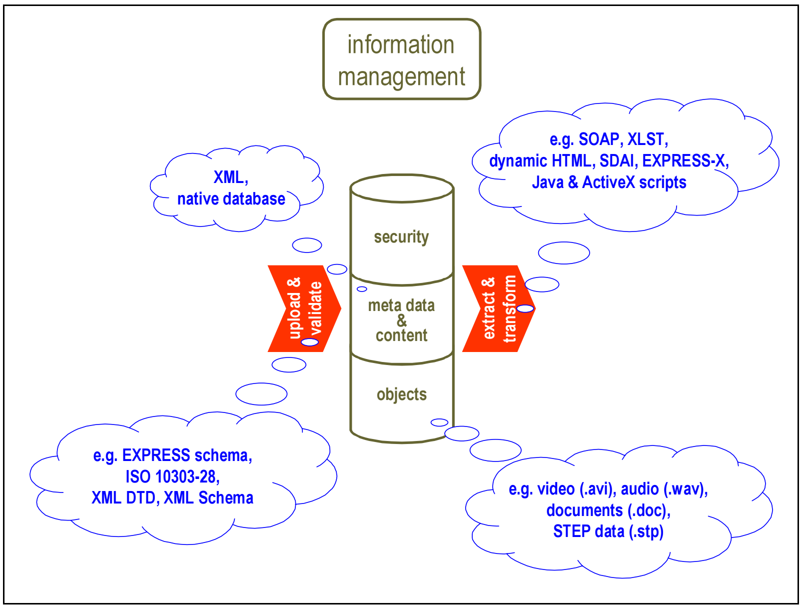


Figure 16 The required data management architecture

In general, the repository should handle both coarse-grained and fine-grained data.

The third and final phase is to deliver data to the various end users [see **Figure 17**]. STEP is relevant to this delivery, especially in the situation where the system is providing access for a user of one CAD application to a model from another, different application. However, XML and, decreasingly, HTML are more likely to be the prevalent technologies because of the availability of cheap browsing tools. A functionality will be necessary to bind the users together in a single environment and, thus, support management of capabilities such as workflow, user profiles and access security.

A shared data environment binds together all the users across the extended enterprise.

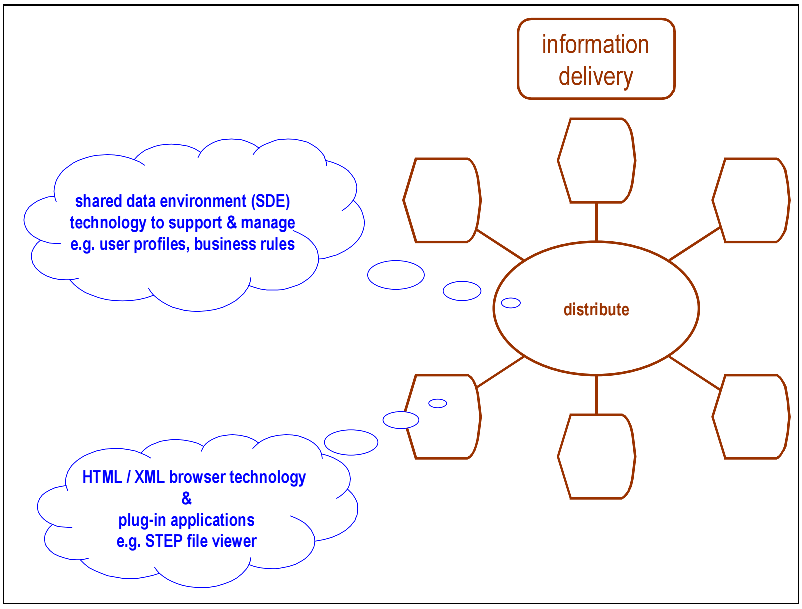


Figure 17 Delivery through a shared data environment

The extended enterprise scenario shows how both XML and STEP technologies are relevant to the broad requirements of product data integration. In fact, for all the major capabilities, both technologies have candidate solutions [see **Table 2**]. Thus, the final design of an enterprise architecture will depend on the balance of various key factors, which characterise the data, software, processes and nature of the enterprise [see **Table 3**].

Table 2 The potential application of STEP and XML for product data integration across the extended enterprise

Capability	**Applicable technology**	
	STEP	**XML**
conceptual modelling	ISO 10303-11 (EXPRESS)	XML Schema
product data representation	ISO 10303-21 (clear text) ISO 10303-28 (XML)	XML
meta-data	ISO 10303-11 (EXPRESS)	XML XML Schema
delivery representation transformation	ISO 10303-14 (EXPRESS-X)	XSLT
user presentation	ISO 10303-21 (clear text) ISO 10303-28 (XML)	XML XSL

Table 3 Key factors in the design of an architecture for product data integration across the extended enterprise

Category	Feature
nature of the data	volume
	rate of change
	scope (breadth) of engineering domains
	complexity of hierarchies (depth)
	granularity of end user views
	requirement for meta-data
software applications for source data	number
	past, current & future utility (e.g. legacy systems)
	rate of change
	software & hardware technology
	available interfaces (e.g. to a STEP format)
nature of the enterprise	number of distinct organisations
	top-level business objectives
	rate of change
	degree of geographic dispersion
	degree of information technology sophistication
	available interconnectivity
	value of target product or products (e.g. ship)
nature of the processes	degree of concurrency in engineering
	rate of change
	configuration management & change control

Faster, better, cheaper forms the fundamental set of business drivers.

Ultimately, in an industry such as ships where the extended enterprises is the *modus operandi* for creating and operating the end product, the fundamental business drivers are the same as for any commercial operation: "faster, better, cheaper". No one-size, fits-all technology is available but the creation of a knowledge-rich environment is the end goal. A coherent set of information technology provides the basis on which the enterprise can seek to minimise the total cost of ownership for the end product.

Conclusions

XML is exciting and now firmly in the mind of the general end user.

STEP versus XML is not a straightforward "winner-takes-all" contest. STEP is a mature, yet ever-developing International Standard. XML is a host of specifications and technologies. W3C controls the specifications but, in the wider community, XML is caught up in a surge of Internet-driven energy.

In the eyes of some observers, XML apparently threatens to swamp STEP because the momentum has delivered the capability to "do XML" at the desk of every end user for a low cost of entry. Popularity and "X appeal" are seemingly compelling motivations to implement XML.

The development of XML continues apace as implementations uncover limitations in the foundational concepts.

In the light of the ever-increasing number of implementations, XML is responding to the revealed limitations in the technology. For instance, although the readability of XML tags is an intuitive mechanism by which the user can attempt to determine the meaning of content, the DTD is neither definitive in controlling typing of data nor particularly rich in terms of definable structure. Thus, the XML community is working on XML Schema.

XML provides powerful support for the implementation of flexible linking mechanisms and the enabling and management of distributed data. Such support underpins a capability to deliver a web of knowledge to end users.

Overall, the XML garden appears to be full of roses. However, in fact, none of these attractions begins to challenge the foundations of STEP. ISO 10303 specifies the architecture to deliver information models that are complete, unambiguous and conformance testable. These conceptual models are independent of implementation technology and, thus, have the longevity to outlast any particular software mechanisms for realising the purposes of the model.

Context for data can be insufficient. The XML tags `<building>` … `<floor_number>` … will not be robust when the American starts counting "1" at ground level in contrast to the British custom, where the first floor is above the ground floor.

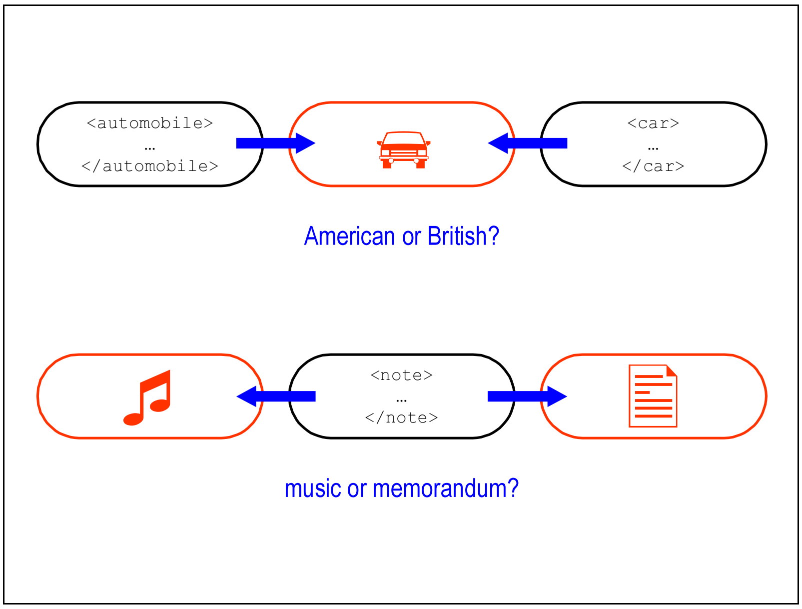


Figure 18 Confusion is just around the corner …

The STEP Application Protocols embody extensive work on capturing the information requirements for activities in various engineering domains. Such information requirements are a necessary basis for product data integration across the enterprise. The core technologies of XML do not achieve the integration of information requirements. In particular, any given pair of Document Type Definitions will possibly suffer in the absence of a common vocabulary [see **Figure 18**]. Ultimately, XML initiatives are necessary to avoid such confusion and to establish the vocabulary that will allow the widespread, consistent automation of information processing. Meanwhile, STEP has already achieved agreement across large areas of product data.

Data exchange using STEP would have prevented the Mars Orbiter disaster.

Returning to the example of the NASA Orbiter [see **Figure 11**], one XML initiative is citing the disaster **[15]** and conducting work to ensure that adequate representation of units is available within XML **[16]**. However, a rigorous information model for units has been available as an International Standard within STEP since 1994 **[17]**. If the NASA teams had exchanged information in accordance with an appropriate STEP Application Protocol then the Orbiter would not have failed to reach Mars safely. As an encoding mechanism, XML can not directly address the issues that cause users and systems developers to mistake the nature of data.

Instead of competition, the capabilities of XML are actually the basis for wider implementation of STEP. Thus, the STEP community has developed Part 28 of the standard. With a capability to represent EXPRESS schemata and conforming instance data in XML, the enterprise is able to build architectures that encompass a broader scope than the contents of ISO 10303. This broader scope includes applications such as workflow, product data management and e-commerce frameworks for the supply chain.

Total cost of ownership is as relevant to information systems as to major asset products.

In the search to bind together the extended enterprise, although the cost of entry is potentially low for XML technologies, this initial cost is not going to drive design decisions because so many other key factors influence the overall architecture. The solution will require a long-term resilience and scalability, whereby the total cost of ownership will depend on the effect and scale of business process and organisational changes.

In summary, STEP versus XML is a false controversy. The two standards have arisen from requirements at opposite ends of the information continuum. The motivation for STEP is the specific support of product engineering through agreed common models of information requirements. XML is a representation mechanism that is rapidly becoming a technology of first choice. In order to achieve effective product data integration across the extended engineering enterprise, STEP and XML are complementary technologies and are both necessary.

References

[01] Hoare, Graydon: "Lex me, Amadeus"
`http://www.venge.net/graydon/markup-abuse.html`

[02] Vaughan, C.; & Ishikawa, Y.: "SC4 industrial data framework", ISO TC184/SC4/WG10 N326, 08 November 2000
`http://wg10step.aticorp.org/Deliverables/Framework/SSID/nov2000/sc4frameworkN326.pdf`

[03] Cover, Robin: "The XML cover pages: Web site index: Expanded outline", 12 December 2000
`http://www.oasis-open.org/cover/siteIndex.html`

[04] "DTD/schema for product data exchange"
`http://xml.erim.org/PDMDTD.htm`

[05] "XML for materials property data"
`http://www.ceramics.nist.gov/matml/matml.htm`

[06] "An industry document standard for metal trading"
`http://www.metal-xml.org/`

[07] "Product Data Markup Language"
`http://www.pdml.org/`

[08] "PDX Standard Group"
`http://www.pdxstandard.org/`

[09] "Universal Commerce Language and Protocol (UCLP)"
`http://www.w3.org/TR/1999/NOTE-uclp-19990120/`

[10] Burkett, William C.: "PDML (Product Data Markup Language): A new paradigm for product data exchange and integration", White Paper, 30 April 1999
`http://www.pdml.org/whitepap.pdf`

[11] St. Laurent, S.: "XML Schemas: Needed rules for XML documents", interChange, June 2000, pages 13 to 17

[12] Rivers-Moore, Daniel: "XML and EXPRESS as schema definition languages", Meta Structures 98 Conference, Montreal, Canada, August 1998
`http://www.rivcom.com/knowledge/papers.htm#xmlandexpress`

[13] "Product data representation and exchange: Implementation methods: XML representation of EXPRESS schemas and data", ISO/PDTS 10303-28, ISO TC184/SC4/WG11 N140, 6 October 2000
`http://www.nist.gov/sc4/wg_qc/wg11/n140/`

[14] Budd, Mike; Harris-Jones, Christopher; & Wells, David: "Repositories and XML: The future of metadata management", Ovum, December 1999

[15] Olken, Frank; & John McCarthy: "Measurement units in XML datatypes", 1999 June 7
`http://pueblo.lbl.gov/~olken/mendel/w3c/xml.schema.wg/units/syntax.htm`

[16] Olken, Frank; & John McCarthy: "What XML schema designers need to know about measurement units", XTech 2000 Presentation
`http://www.oasis-open.org/cover/olkenMeasurementAbstract.html`

[17] ISO 10303-41: "Industrial automation systems and integration -- Product data representation and exchange -- Part 45: Integrated generic resource: Fundamentals of product description and support"

[18] Kemmerer, Sharon J.: "STEP: The grand experience", NIST Special Publication 939, Manufacturing Engineering Laboratory, National Institute for Standards and Technology, Gaithersburg, July 1999
`http://www.mel.nist.gov/msidlibrary/doc/stepbook.pdf`

[19] Zisman, A.: "An overview of XML", Computing & Control Engineering Journal, August 2000, pages 165 to 167

[20] ISO 10744: "Information technology -- Hypermedia/Time-based Structuring Language (HyTime)"

[21] ISO 10179: "Information technology -- Processing languages -- Document Style Semantics and Specification Language (DSSSL)"

[22] Connolly, Dan: "Extensible Markup Language (XML) activity"
`http://www.w3.ord/XML/Activity`

[23] ISO/CD 10303-14: "Industrial automation systems and integration -- Product data representation and exchange -- Description methods: The EXPRESS-X Language reference manual"

Annex A STEP

ISO 10303 already consists of 6781 pages[2] in the catalogue of the International Organization for Standardization (ISO). At least the same number of pages again is ready for publication as International Standards and many more projects are underway. In 1999, ISO 10303 and two closely related standards were more than 10% of the ISO output by number of pages.

ISO/TC184/SC4 first met on 11 July, 1984 and resolved to create "a standard which enables the capture of information comprising a computerized product model in a neutral form without loss of completeness and integrity, throughout the lifecycle of the product" **[18]**. ISO 10303 has been the primary result of this resolution.

In a way that was not part of the expectations of initial participants in development, ISO 10303 has become an extensive architecture that currently consists of 48 parts at International Standard level and a further 27 parts in the closing drafting stages. Meanwhile, after sixteen years, active development still continues towards many other proposed parts. The development of any individual part will take up to seven years and even longer if enough participating member countries re-confirm their commitment to the work item.

The fundamental objective of ISO 10303 is to enable three core capabilities: data exchange; data sharing; and data archiving [see **Figure 19**]. These capabilities are between discrete information systems and without loss of meaning.

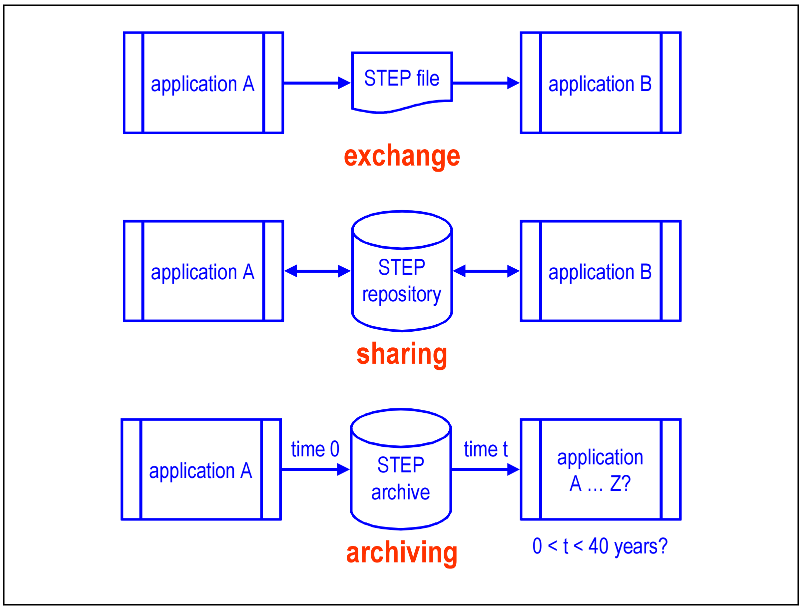


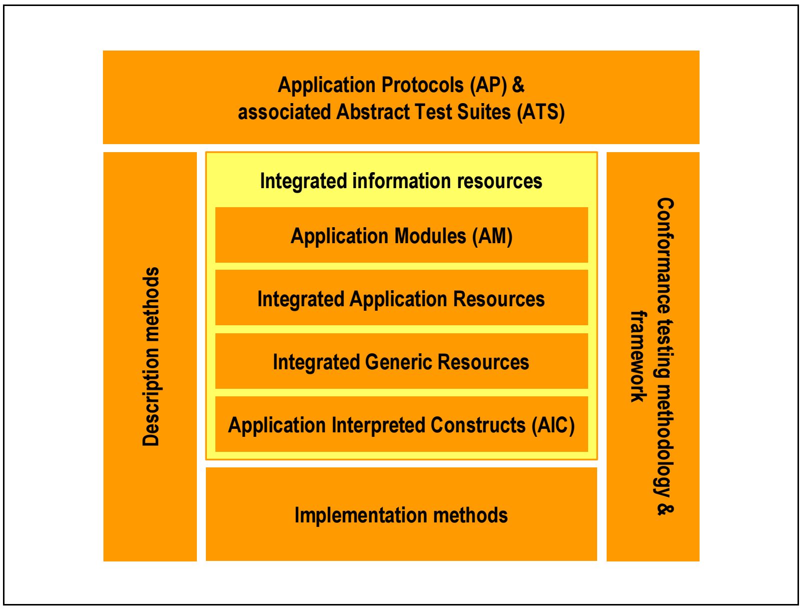
Figure 19 Usage scenarios for STEP

The complete architecture of ISO 10303 consists of several different types of component [see **Figure 20** to **Figure 22**]. The Application Protocols are of direct interest to end users. An Application Protocol is the basis by which to enable a STEP capability for end-user software and to deliver business benefits to engineering enterprises. For example, the supplier to multiple manufacturers can operate a single CAD system, even if the manufacturers have different systems.

[2] As at 26 October, 2000.

Each Application Protocol specifies the industrial information requirements for a particular engineering domain. The domain is either in terms of an industry sector or a technical discipline. The current domains within ISO 10303 include:

- industry sectors such as automotive, electrical and electronic, ships, process plant and furniture;
- technical disciplines such as engineering drawing and design, manufacturing, systems engineering and technical data packaging.



Note: for further details see **Figure 21** and **Figure 22**.

Figure 20 The architecture of ISO 10303 (STEP)

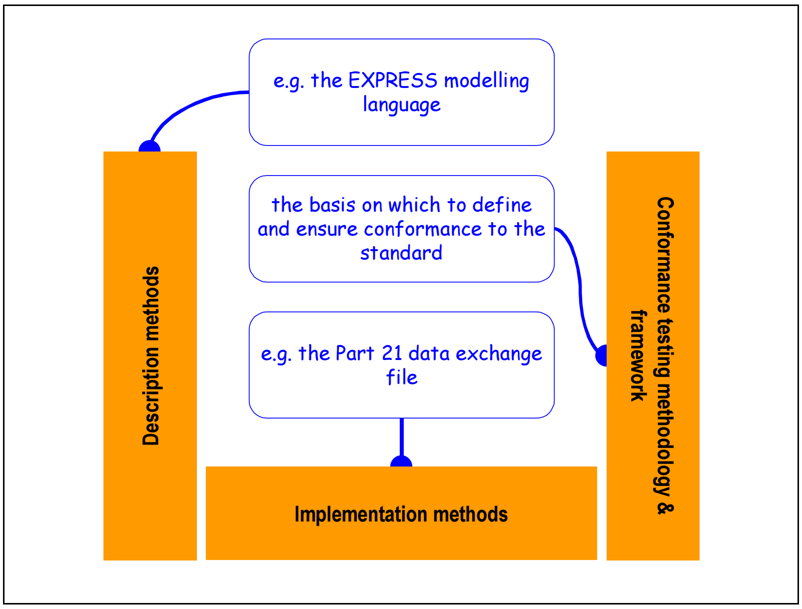


Figure 21 The foundational components of ISO 10303

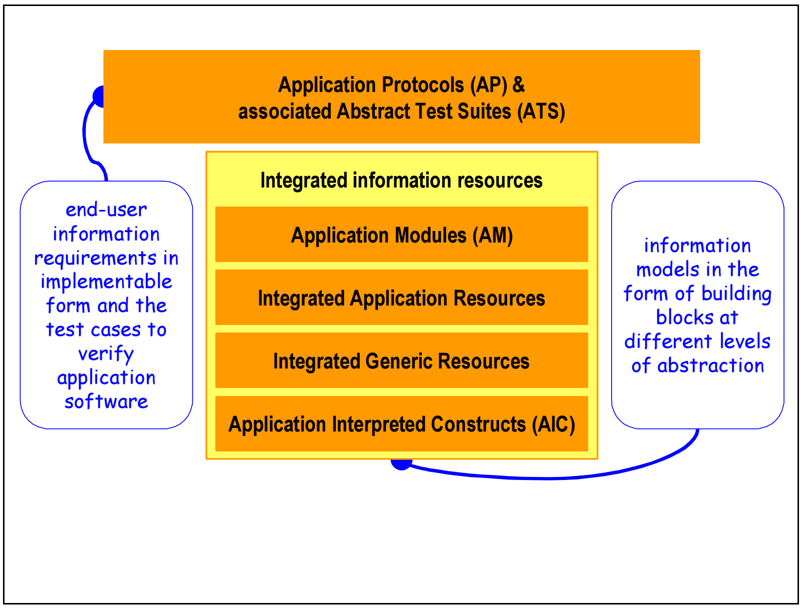


Figure 22 The information models of ISO 10303

Of all the other components of ISO 10303, perhaps the most widely recognised is EXPRESS, which is Part 11 of the standard and is a data specification language. ISO/TC184/SC4 has used EXPRESS in creating both STEP and the sibling standards[3]. However, other communities have also used the language for modelling information requirements within geographic, financial and human genome domains.

EXPRESS provides the capability to represent the information requirements of STEP as a formal conceptual model. Without going to the depth of detail in the information models of the actual standard, a single example can demonstrate the fundamental features of an EXPRESS schema [see **Figure 23**]. A complete model will perhaps consist of multiple, inter-linked schemata.

[3] Three product data standards: ISO 13584 “Parts library” (PLib); ISO 15531 “Industrial manufacturing management data” (MANDATE); and ISO 15926 “Integration of life-cycle data for oil and gas production facilities” (Oil & Gas).

```
SCHEMA example_part_schema;

ENTITY manufacturer;
       name : STRING;
       headquarters : STRING;
END_ENTITY;

ENTITY part;
       number : STRING;
       description : OPTIONAL STRING;
       release_date : date;
       produced_by : manufacturer;
       UNIQUE
       unq : produced_by, number;
END_ENTITY;

ENTITY date;
       day : INTEGER;
       month : INTEGER;
       year : INTEGER;
       WHERE
       days_in_month : { 1 <= day <= 31 };
END_TYPE;

END_SCHEMA;
```

Note: see **Figure 24** for a graphical form of this schema.

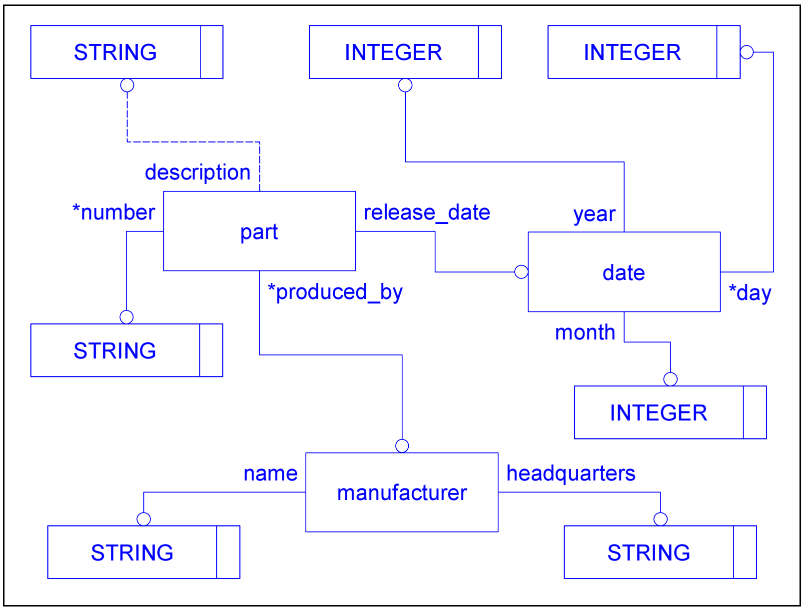
Figure 23 An example of an EXPRESS schema

In order to simplify the presentation of complex schemata for the purposes of human review, ISO 10303-11 also specifies EXPRESS-G, which is a graphical form of the base language [see **Figure 24**]. This graphical form provides a useful overview of the schema but is not a complete specification of the model. EXPRESS-G is unable to represent the precise details of rules and, instead, a "*" indicates any attribute that is subject to the rule. In the example, a `UNIQUE` rule constrains the combination of the `produced_by` and `number` attributes of the `part` entity and a `WHERE` rule constrains the `days_in_month` attribute.

Once a schema is available, the user is able to create a set of instance data that conform to that schema. The STEP architecture includes various specific forms of Implementation Method but the primary distinction is between an:

- exchange file; and
- access interface.

The first generation of STEP-capable applications has relied on two key documents: ISO 10303-21 and ISO 10303-22. ISO 10303-21 specifies the "Clear text encoding of the exchange structure". Informally, the STEP community talks about the "Part 21 file" [see **Figure 25** for a set of instance data that conform to the example schema]. The access interface has both a generic structure and bindings in specific languages. ISO 10303-22 specifies the generic structure, the "Standard data access interface" and the informal name is "SDAI".



Note: this graphical form corresponds to the schema in **Figure 23**.

Figure 24 The example EXPRESS schema in graphical form

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('THIS FILE IS A SMALL EXAMPLE'),'3;1');
FILE_NAME('EXAMPLE PART 21 FILE','2000-12-01T17:13:21',
('FRED SMITH','JONES BROTHERS','ANYTOWN UK'),
('JONES BROTHERS','ANYTOWN UK'),
'HANDCRANK 2.0',
'MINDERP 4.2A',
'APPROVED BY JOE BLOGGS, JONES BROTHERS, ANYTOWN');
FILE_SCHEMA(('EXAMPLE_PART_SCHEMA'));
ENDSEC;
DATA;
#1=MANUFACTURER('Smith & Jones Limited','Leeds, UK');
#2=DATE(19,1,1998);
#3=DATE(23,11,1997);
#4=PART('56-7890','Widget',#2,#1);
#5=PART('9A-431217890','Widget flange',#3,#1);
ENDSEC;
END-ISO-10303-21;
```

meta-data for the file

actual instance data

Note: the schema in **Figure 23** specifies the entity instances in this file.

Figure 25 An example of a STEP Part 21 file

Annex B XML

General features of the language

Arguably, in terms of both speed of development and impact, XML is a world removed from STEP and is a concrete example of the modern phenomenon of "Internet time". The W3C is responsible for the Recommendation that specifies XML, having formed as the protector of World Wide Web technologies when the Internet underwent explosive growth during the 1990s **[19]**. Work began on XML in June 1996 and the XML 1.0 Recommendation became available in February 1998. Since that date, XML has captured a general attention that places the letters "X" or "ML" as a seemingly vital part of any to-be-approved information technology project.

XML is a clear, simple concept in terms of the base language and, despite the hype that suggests the language is a revolutionary new development, follows a long understood approach to structured representation of data [see **Figure 26**]. The direct parent of XML is Standard Generalized Markup Language (SGML), which is an International Standard (ISO 8879). HyperText Markup Language (HTML) was the first widely successful attempt to specialise SGML for the Internet. However, HTML is only truly appropriate to choose the layout by which to present data and does not embody any of the SGML capabilities in respect of user-determined structuring of content. In simple terms, HTML only allows the user to apply existing tags (e.g. `<table>…</table>`) that describe the presentation of documents. In contrast, XML provides the capability to create new tags that, through the name, can indicate the underlying meaning of the data.

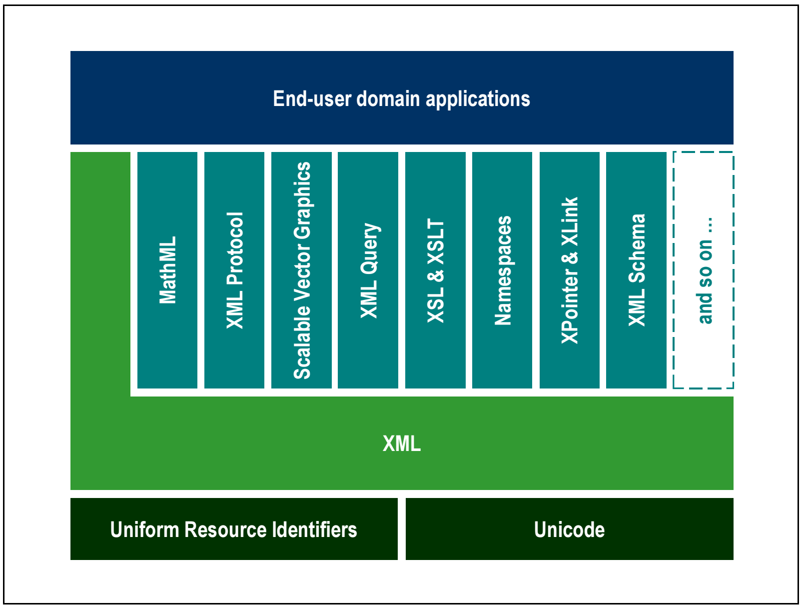
```
<manufacturer id = "ABCD">
       <manu_name>Smith & Jones Limited</manu_name>
       <manu_headquarters>Leeds, UK</manu_headquarters>
</manufacturer>

<part id = "1234">
       <part_number>56-7890</part_number>
       <part_description>Widget</part_description>
       <part_rel_date>19 January 1998</part_rel_date>
       <part.manufacturer>
              <manufacturer_ref refid = "ABCD"/>
       </part.manufacturer>
</part>

<part id = "1235">
       <part_number>9A-431217890</part_number>
       <part_description>Widget flange</part_description>
       <part_rel_date>23 November 1997</part_rel_date>
       <part.manufacturer>
              <manufacturer_ref refid = "ABCD"/>
       </part.manufacturer>
</part>
```

Figure 26 An example of XML markup

Further to the simple concepts of the base language, the XML community has begun to deliver a host of technologies that extend the core representation capability [see **Figure 27**]. Thus, using the label "XML" refers potentially to any one of four items:

- the base language, which now exists as a Second Edition of the 1.0 Recommendation;

- extensions to the language, such as XML Inclusions (XInclude), XML Information Set and XML Fragment Interchange. These extensions are all interoperable;
- support and implementation technologies and standards, such as Extensible Stylesheet Language (XSL), XML Query and XML Schema;
- industry and business initiatives to develop applications of XML, such as Mathematical Markup Language (MathML) and Metal XML.



Note: Uniform Resource Identifiers is the addressing mechanism that binds together the World Wide Web and Unicode is the character set that supports the vast majority of characters from the written languages of the world.

Figure 27 XML: a language and a host of related technologies

Some of the XML capability and extensions encompass the content of International Standards that take SGML beyond an initial core: ISO 10744 **[20]**; and ISO 10179 **[21]**.

W3C operates as a self-regulating community bound by a shared set of interests. The Consortium has established formal procedures for the development of standards, a Recommendation being the final product of the development process. These procedures are applicable to any one of the four items above, although many of the initiatives will operate independently and outside of the direct control of W3C. Without doubt, the communication opportunities of the Internet have allowed groups of diverse individuals to form and rapidly create new standards.

XML developed out of SGML on the basis of a new generation of requirements that have arisen in the Internet era. SGML has achieved significant penetration in the publishing industry but provides a large overhead of functionality that is unnecessary for the majority of applications. The base language of XML is a more reasonable balance of power and weight. The most obvious example of this balance is the changed status of the Document Type Definition (DTD), which is the mechanism to identify the necessary structure of the data, including aspects such as the valid names of tags.

Access to a DTD is a mandatory requirement for the conformant processing of data for SGML. However, complex structures will require large Document Type Definitions and the exchange and processing of the DTD will often be a significant burden within the overall operation to handle the data content. The DTD is still the mechanism by which to ensure valid markup within XML. However, the Document Type Definition is optional and the expression “valid markup” contrasts with “well-formed markup”, which implies that the XML content has no syntax errors but does not indicate whether or not the structure is consistent with any particular DTD. A user of XML can conform to the standard without using the DTD and, thus, avoid the processing burden.

The initial formulation of XML has proven to be incomplete in respect of the requirements of actual use of the language. This incompleteness has been the driver behind the extensions and implementation technologies that provide several major classes of functionality. These classes include:

- rich control of structure through, for example, the use of more extensive and extensible data types including string, boolean and integer and so on;
- powerful linking to provide, for example, bi-directional links;
- management of distributed data through, for example, namespaces that allow for specification of the scope within which identities are applicable;
- transformation to allow, for example, the same source XML file to be the basis for the different views of both the finance department and the engineering team.

The details of all the applicable technologies are not relevant to this White Paper. However, a summary of several such technologies is a useful starting point [see **Table 4**]. The following sections provide further details of how these technologies enable the above classes of functionality **[22]**.

Rich control of structure

XML inherited the DTD as the mechanism by which to ensure valid markup but, in comparison with the typical capabilities required for information management purposes, the DTD has limitations in the flexibility and power of specifying the structure and data types of content. Information systems analysts view these limitations as a retrograde step when considering the deployment of XML technologies.

Several proposed mechanisms are available to extend the capability of XML in respect of rich control of structure **[11]**. These mechanisms include: Schematron; Regular Language description for XML (RELAX); and XML Schema.

Rich control of structure ensures a greater control of data integrity, for example by forcing years to be integer values and all physical measurements to have an identified, associated unit. Such integrity ensures that computer applications can inter-operate through the exchange and sharing of data. Furthermore, the true semantic intent of the data is more apparent to programmers who interpret the structures during development of these applications.

Table 4 A summary guide to some key XML technologies

Technology		Functionality			
Name & reference	**Description**	**Structure**	**Link**	**Distribute**	**Transform**
Canonical XML http://www.w3.org/TR/xml-c14n	Method for ensuring the logical equivalence of XML documents despite any physical differences.			✓	
Extensible Stylesheet Language (XSL) http://www.w3.org/Style/XSL/	Language and vocabulary for format transformation of XML documents.				✓
Namespaces http://www.w3.org/TR/REC-xml-names/	Method for qualifying element and attribute names in XML documents.			✓	
Resource Description Framework (RDF) http://www.w3.org/RDF/	Integration of many forms of meta-data for WWW content.			✓	
Regular Language description for XML (RELAX) http://www.xml.gr.jp/relax/	Method for specifying the structure and meaning of XML documents.	✓			
Schematron http://www.ascc.net/xml/resource/schematron/schematron.html	Method for specifying the structure of XML documents.	✓			
Simple Object Access Protocol (SOAP) http://www.w3.org/TR/SOAP/	Method for message exchange in distributed environments.			✓	
XML Linking Language (XLink) http://www.w3.org/XML/Linking.html	Method for defining links in XML documents between resources.		✓		
XML Base http://www.w3.org/TR/xmlbase/	Method for defining a base address applicable to all references in an XML document.		✓		
XML Fragment Interchange http://www.w3.org/TR/WD-xml-fragment	Method by which to allow access to fragments of XML documents.			✓	
XML Information Set http://www.w3.org/TR/xml-infoset/	Definition of the component features and associated properties of an XML document.			✓	

Technology		Functionality			
Name & reference	**Description**	**Structure**	**Link**	**Distribute**	**Transform**
XML Inclusions (XInclude) http://www.w3.org/TR/xinclude/	Method for combination of XML documents to create a coherent whole.			✓	
XML Protocol http://www.w3.org/2000/xp/	Technologies for communication between peers in a distributed environment.			✓	
XML Query http://www.w3.org/XML/Query	Facilities to extract data from XML documents.			✓	
XML Schema http://www.w3.org/XML/Schema	Method for specifying the structure and meaning of XML documents.	✓			
XML Signature http://www.w3.org/TR/xmldsig-core/	Integrity, authentication and authority checking for XML documents.			✓	
XML Path Language (XPath) http://www.w3.org/TR/xpath	Language for addressing parts of an XML document.		✓		
XML Pointer Language (XPointer) http://www.w3.org/XML/Linking.html	Language for identifying fragments of XML documents on the WWW.		✓		
XSL Transformations (XSLT) http://www.w3.org/TR/xslt.html	Language for transformation of XML documents.				✓

Powerful linking

The fundamental component of the World Wide Web is the link. However, as any even short-time user of the Internet will also testify, the links are prone to lack content, breakdown and become obsolete. XML requires additional forms of link technology in order to ensure a broader range of more robust capability for the language.

The features of powerful linking include: bi-directional linking; linking to multiple destinations; roles and descriptions for links; linking relative to a base address; linking to read-only documents; databases of links; linking to unmarked destinations; linking to fine-grained destinations; and identification of destinations in terms of document structure. These features are becoming available in XML through: XLink; XPath; XPointer; and XML Base.

In addition to general robustness, powerful linking ensures: users can navigate through a rich and vast network of data; non-XML formats of data will remain in the particular format and yet serve a seamless role within the XML data content; and creation of new data does not require duplication of existing source data. Links are the foundation for better connectivity of distributed data and, thus, increased effectiveness within the extended enterprise.

Management of distributed data

As linking technology encourages the increased deployment of distributed data, management of the data becomes a significant challenge. Given the nature of the World Wide Web, the necessary range of capabilities and technologies is broad. This range includes: coherence of identity through Namespaces; enabling and managing meta-data activities through Resource Description Framework (RDF); searching and retrieval of data through XML Query; management of content granularity through XML Inclusions, XML Information Set and XML Fragment Interchange; identification of authorship through XML Signature and Canonical XML; messaging and application interaction through XML Protocol and Simple Object Access Protocol (SOAP).

Transformation

Although XML provides a capability to specify the structure of data, any given structure is not likely to be applicable to all possible uses. When managing data in the extended enterprise, the primary requirement is for integration structures that unify data. However, the extended enterprise will include wide variation across the different end users of data. These end users will require data in representations that are specific to and differ according to the type of user.

Extensible Stylesheet Language (XSL) provides the basic capability to specify the format of presentation independently of the content in any particular XML file. XSL Transformations (XLST) builds on this capability to provide a mechanism that has great flexibility in making the transformation from one form of the XML content to a different form.

The STEP community has also recognised the diversity that is likely to exist in schemata across the extended enterprise. EXPRESS-X **[23]** is an extension of the base EXPRESS language and provides a transformation capability for ISO 10303.

The user can specify in EXPRESS-X a mapping that will serve as a bridge between two EXPRESS schemata. Once the bridge is in place, the computer can transform data that conforms to one schema so as to conform to the second schema.

Although transformation is a vital requirement, one unavoidable limitation does exist and is applicable to both XSLT and EXPRESS-X. The fundamental nature of the transformation process is such that the computer can not automatically reverse the process. Instead, if a bi-directional capability is necessary then the user must explicitly define two different mappings.

Contact details

Tony Fry, EMSA Chairman

Warships Support Agency,
MoD UK,
Fox Hill,
Bath, UK – BA1 5AB

Telephone: +44-1225-884162
Facsimile: +44-1225-885725
e-mail: tf@navils.demon.co.uk